```
import pandas as pd

df_flights = pd.read_csv('/content/flights.csv')
df_flights.head()
```

	Year	Month	DayofMonth	DayOfWeek	Carrier	OriginAirportID	OriginAirportName	OriginCity	OriginState	DestAirportID
0	2013	9	16	1	DL	15304	Tampa International	Tampa	FL	12478
1	2013	9	23	1	WN	14122	Pittsburgh International	Pittsburgh	PA	13232
2	2013	9	7	6	AS	14747	Seattle/Tacoma International	Seattle	WA	11278
3	2013	7	22	1	OO	13930	Chicago O'Hare International	Chicago	IL	11042
4	2013	5	16	4	DL	13931	Norfolk International	Norfolk	VA	10397

```
df_flights.isnull().sum()
```

```
Year                 0
Month                0
DayofMonth           0
DayOfWeek            0
Carrier              0
OriginAirportID      0
OriginAirportName    0
OriginCity           0
OriginState          0
DestAirportID        0
DestAirportName      0
DestCity             1
DestState            1
CRSDepTime           1
DepDelay             1
DepDel15            88
CRSArrTime           1
ArrDelay             1
ArrDel15             1
Cancelled            1
dtype: int64
```

```
df_flights[df_flights.isnull().any(axis=1)][['DepDelay','DepDel15']]
```

	DepDelay	DepDel15
171	0.0	NaN
359	0.0	NaN
429	0.0	NaN
545	0.0	NaN
554	0.0	NaN
...	...	...
7273	0.0	NaN
7436	0.0	NaN
7686	0.0	NaN
7787	0.0	NaN
7963	NaN	NaN

88 rows × 2 columns

```
df_flights[df_flights.isnull().any(axis=1)].DepDelay.describe()
```

```
count    87.0
mean      0.0
std       0.0
min       0.0
25%       0.0
50%       0.0
75%       0.0
```

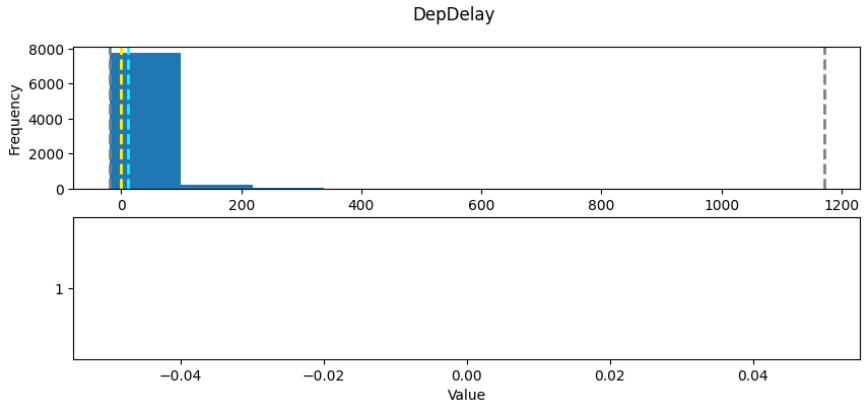
```
max       0.0
Name: DepDelay, dtype: float64
```

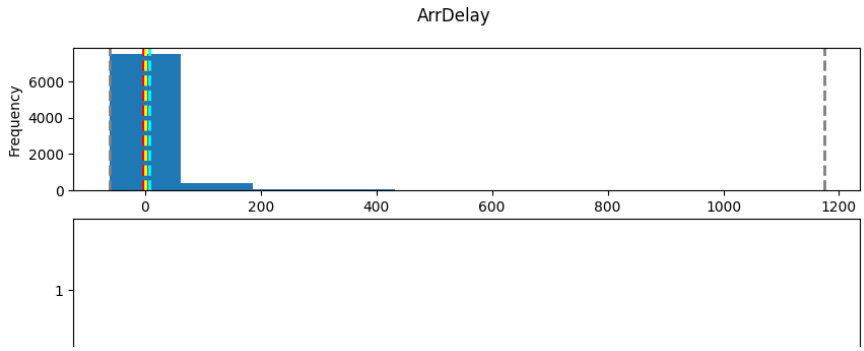
```
df_flights.DepDel15 = df_flights.DepDel15.fillna(0)
df_flights.isnull().sum()
```

```
Year                0
Month               0
DayofMonth          0
DayOfWeek           0
Carrier             0
OriginAirportID     0
OriginAirportName   0
OriginCity          0
OriginState         0
DestAirportID       0
DestAirportName     0
DestCity            1
DestState           1
CRSDepTime          1
DepDelay            1
DepDel15            0
CRSArrTime          1
ArrDelay            1
ArrDel15            1
Cancelled           1
dtype: int64
```

```
# Function to show summary stats and distribution for a column
def show_distribution(var_data):
    from matplotlib import pyplot as plt

    # Get statistics
    min_val = var_data.min()
    max_val = var_data.max()
    mean_val = var_data.mean()
    med_val = var_data.median()
    mod_val = var_data.mode()[0]

    print(var_data.name,'\nMinimum:{:.2f}\nMean:{:.2f}\nMedian:{:.2f}\nMode:{:.2f}\nMaximum:{:.2f}\n'.format(min_val,
                                                                                                          mean_val,
                                                                                                          med_val,
                                                                                                          mod_val,
                                                                                                          max_val))

    # Create a figure for 2 subplots (2 rows, 1 column)
    fig, ax = plt.subplots(2, 1, figsize = (10,4))

    # Plot the histogram
    ax[0].hist(var_data)
    ax[0].set_ylabel('Frequency')

    # Add lines for the mean, median, and mode
    ax[0].axvline(x=min_val, color = 'gray', linestyle='dashed', linewidth = 2)
    ax[0].axvline(x=mean_val, color = 'cyan', linestyle='dashed', linewidth = 2)
    ax[0].axvline(x=med_val, color = 'red', linestyle='dashed', linewidth = 2)
    ax[0].axvline(x=mod_val, color = 'yellow', linestyle='dashed', linewidth = 2)
    ax[0].axvline(x=max_val, color = 'gray', linestyle='dashed', linewidth = 2)

    # Plot the boxplot
    ax[1].boxplot(var_data, vert=False)
    ax[1].set_xlabel('Value')

    # Add a title to the Figure
    fig.suptitle(var_data.name)

    # Show the figure
    fig.show()

# Call the function for each delay field
delayFields = ['DepDelay','ArrDelay']
for col in delayFields:
    show_distribution(df_flights[col])
```

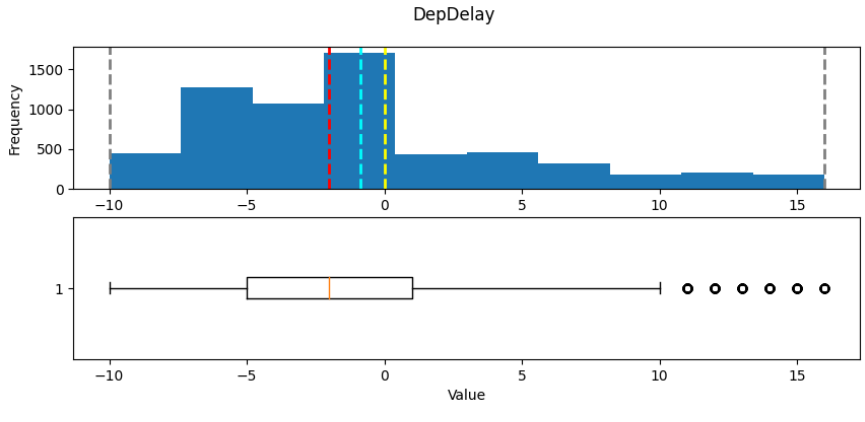
```
DepDelay
Minimum:-20.00
Mean:10.42
Median:-1.00
Mode:0.00
Maximum:1172.00

ArrDelay
Minimum:-62.00
Mean:6.44
Median:-3.00
Mode:0.00
Maximum:1175.00
```






```
# Trim outliers for ArrDelay based on 1% and 90% percentiles
ArrDelay_01pcntile = df_flights.ArrDelay.quantile(0.01)
ArrDelay_90pcntile = df_flights.ArrDelay.quantile(0.90)
df_flights = df_flights[df_flights.ArrDelay < ArrDelay_90pcntile]
df_flights = df_flights[df_flights.ArrDelay > ArrDelay_01pcntile]

# Trim outliers for DepDelay based on 1% and 90% percentiles
DepDelay_01pcntile = df_flights.DepDelay.quantile(0.01)
DepDelay_90pcntile = df_flights.DepDelay.quantile(0.90)
df_flights = df_flights[df_flights.DepDelay < DepDelay_90pcntile]
df_flights = df_flights[df_flights.DepDelay > DepDelay_01pcntile]

# View the revised distributions
for col in delayFields:
    show_distribution(df_flights[col])
```

```
DepDelay
Minimum:-10.00
Mean:-0.88
Median:-2.00
Mode:0.00
Maximum:16.00

ArrDelay
Minimum:-32.00
Mean:-5.10
Median:-6.00
Mode:0.00
Maximum:35.00
```



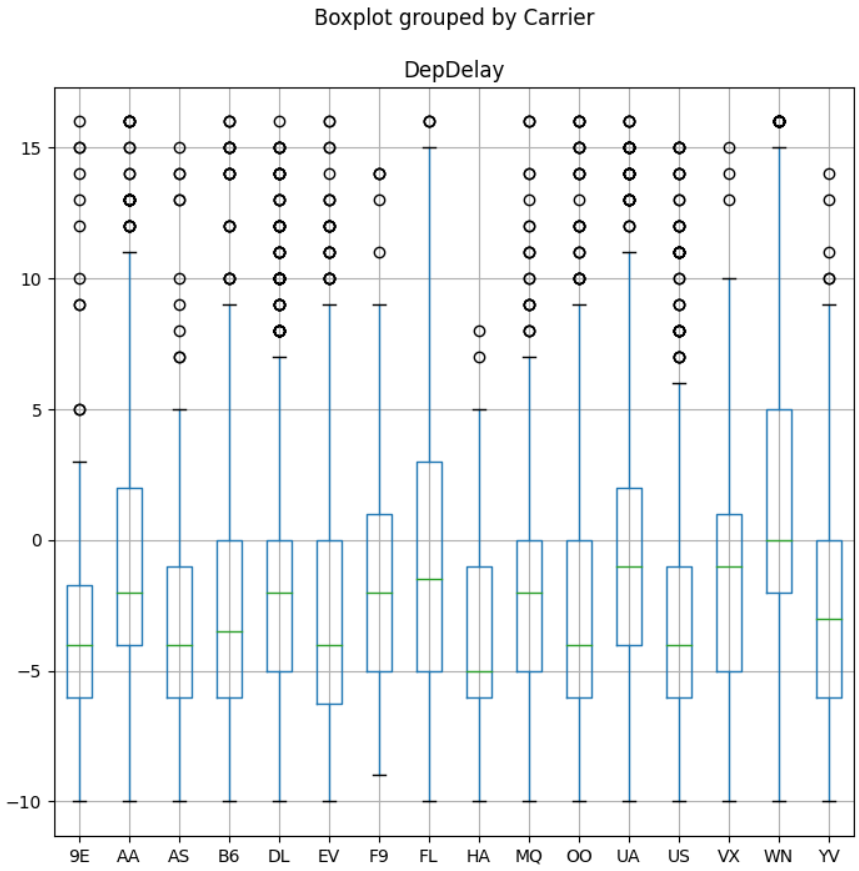

```
df_flights.describe()
```

	Year	Month	DayofMonth	DayOfWeek	OriginAirportID	DestAirportID	CRSDepTime	DepDelay	DepDel15	CRSArr
count	6240.0	6240.000000	6240.000000	6240.000000	6240.000000	6240.000000	6240.000000	6240.000000	6240.000000	6240.00
mean	2013.0	6.991987	15.708974	3.921795	12752.336699	12727.852885	1284.576282	-0.881891	0.016506	1457.70
std	0.0	2.003987	8.842494	2.005835	1511.446400	1501.361465	470.351131	5.596320	0.127423	492.51
min	2013.0	4.000000	1.000000	1.000000	10140.000000	10140.000000	20.000000	-10.000000	0.000000	1.00
25%	2013.0	5.000000	8.000000	2.000000	11292.000000	11292.000000	851.000000	-5.000000	0.000000	1050.00
50%	2013.0	7.000000	16.000000	4.000000	12892.000000	12892.000000	1239.000000	-2.000000	0.000000	1442.00
75%	2013.0	9.000000	23.000000	6.000000	14100.000000	14057.000000	1655.000000	1.000000	0.000000	1845.00
max	2013.0	10.000000	31.000000	7.000000	15376.000000	15376.000000	2359.000000	16.000000	1.000000	2359.00

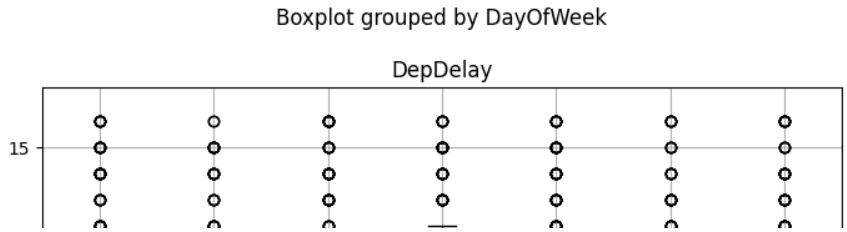
```
#mean departure and arrival delays

df_flights[delayFields].mean()
```

```
DepDelay   -0.881891
ArrDelay   -5.098718
dtype: float64
```

```
#How do the carriers compare in terms of arrival delay performance?
for col in delayFields:
    df_flights.boxplot(column=col, by='Carrier', figsize=(8,8))
```



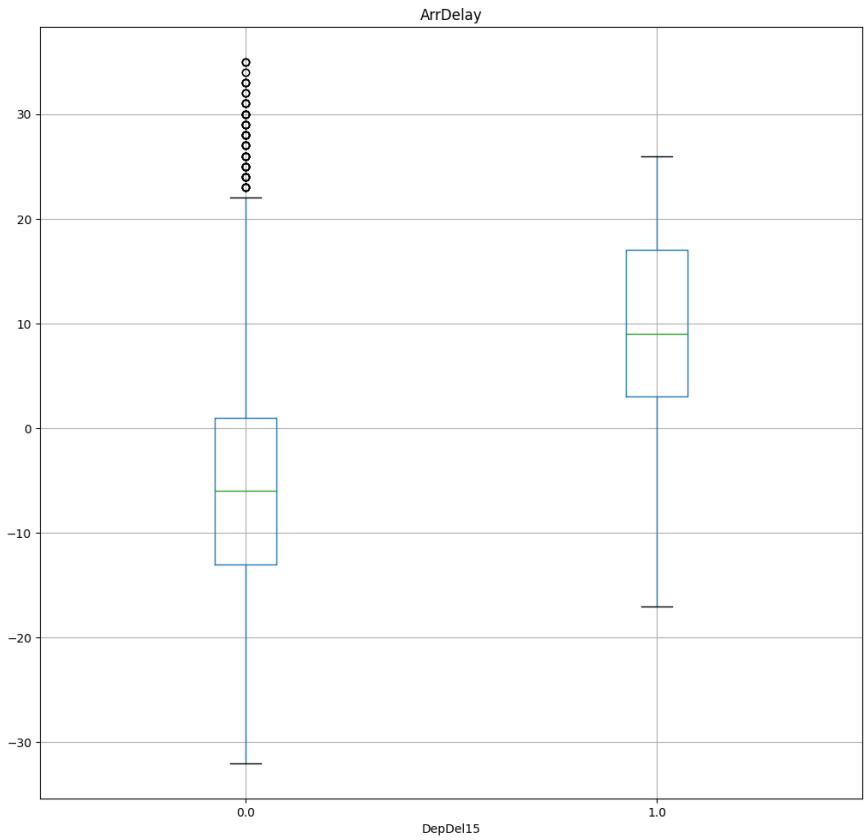

```
#Are some days of the week more prone to arrival days than others?
for col in delayFields:
    df_flights.boxplot(column=col, by='DayOfWeek', figsize=(8,8))
```




```
#Which departure airport has the highest average departure delay?
departure_airport_group = df_flights.groupby(df_flights.OriginAirportName)

mean_departure_delays = pd.DataFrame(departure_airport_group['DepDelay'].mean()).sort_values('DepDelay', ascending=False)
mean_departure_delays.plot(kind = "bar", figsize=(12,12))
mean_departure_delays
```

```
df_flights.boxplot(column='ArrDelay', by='DepDel15', figsize=(12,12))
```

```
<Axes: title={'center': 'ArrDelay'}, xlabel='DepDel15'>
```




```
#Which route (from origin airport to destination airport) has the most late arrivals?
# Add a routes column
routes  = pd.Series(df_flights['OriginAirportName'] + ' > ' + df_flights['DestAirportName'])
df_flights = pd.concat([df_flights, routes.rename("Route")], axis=1)

# Group by routes
route_group = df_flights.groupby(df_flights.Route)
pd.DataFrame(route_group['ArrDel15'].sum()).sort_values('ArrDel15', ascending=False)
```

Route	ArrDel15
Newark Liberty International > San Francisco International	3.0
Baltimore/Washington International Thurgood Marshall > Orlando International	3.0
Dallas/Fort Worth International > San Antonio International	3.0
San Francisco International > Los Angeles International	3.0
Kahului Airport > Honolulu International	3.0
...	...
Jacksonville International > Nashville International	0.0
Jacksonville International > Miami International	0.0
Jacksonville International > Luis Munoz Marin International	0.0
Jacksonville International > John F. Kennedy International	0.0
William P Hobby > Will Rogers World	0.0

1824 rows × 1 columns

```
#Which route has the highest average arrival delay?
pd.DataFrame(route_group['ArrDelay'].mean()).sort_values('ArrDelay', ascending=False)
```

Route	ArrDelay
Raleigh-Durham International > William P Hobby	34.0
Miami International > Denver International	33.0
Orlando International > Kansas City International	31.0
Port Columbus International > Minneapolis-St Paul International	30.0
Ronald Reagan Washington National > Nashville International	29.0
...	...
Newark Liberty International > Seattle/Tacoma International	-31.0
Chicago O'Hare International > Sacramento International	-31.0
John F. Kennedy International > Long Beach Airport	-32.0
John F. Kennedy International > San Antonio International	-32.0
Austin - Bergstrom International > Washington Dulles International	-32.0

1824 rows × 1 columns